

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method for reducing the recovery time after a failure, comprising the steps of:

maintaining storing a checkpoint value that indicates which records of a plurality of records have to be processed after the failure, wherein the plurality of records indicate changes for a plurality of data blocks; and writing changes from volatile memory to nonvolatile memory to advance the checkpoint value based on a user-specified value that corresponds to how much work will be required during a redo phase of recovery.
2. (Original) The method of Claim 1, further comprising the step of:

maintaining, in volatile memory, one or more sorted buffer queues, wherein each sorted buffer queue includes queue entries that are inserted into said sorted buffer queue based on an index value associated with said queue entry, wherein each queue entry reflects a change to a data block of the plurality of data blocks.
3. (Previously presented) The method of Claim 2, wherein the one or more sorted buffer queues are one or more circular sorted buffer queues, and wherein a modulus operation is used to identify the index value associated with each circular sorted buffer queue entry when inserting a queue entry into the circular sorted buffer queue.
4. (Original) The method of Claim 2, further comprising:

maintaining a count of the queue entries in each of the one or more sorted buffer queues.
5. (Previously Presented) The method of Claim 2, wherein each of the one or more sorted buffer queues is associated with a target number of queue entries, and the method further comprises:

determining a count of the queue entries in each of the one or more sorted buffer queues; and

if the count of the queue entries in a particular sorted buffer queue of the one or more sorted buffer queues is greater than target number of queue entries associated with the particular sorted buffer queue, then writing changes to a database to reduce the number of queue entries in the particular sorted buffer queue to the target number of queue entries associated with the particular sorted buffer queue.

6. (Original) The method of Claim 2, wherein the step of updating the checkpoint value comprises:
updating the checkpoint value to equal a byte offset in a redo log associated with the queue entry in the one or more sorted buffer queues that is associated with the least recently modified buffer in any queue entry in the one or more sorted buffer queues.
7. (Original) The method of Claim 1, further comprising the step of:
maintaining, in volatile memory, one or more partially sorted buffer queues, wherein each partially sorted buffer queue includes queue entries that are inserted into said partially sorted buffer queue based on an index value associated with said queue entry,
wherein each queue entry reflects a change to a data block of the plurality of data blocks.
8. (Previously presented) The method of Claim 1, wherein the checkpoint value comprises a byte offset to an identified redo log file.
9. (Currently Amended) A method for controlling an amount of time that is needed to recover after the occurrence of a database system failure, the method comprising the steps of:
maintaining storing a checkpoint value that indicates which records of a plurality of records have to be processed after the failure, wherein the plurality of records indicate changes for a plurality of data blocks;

determining a required recovery time, wherein the required recovery time indicates a maximum length of time that is to be allowed for recovering after said database system failure; and

writing changes from volatile memory to nonvolatile memory to advance the checkpoint value based on the maximum number of data block reads that can be performed in the required recovery time.

10. (Currently Amended) A computer-readable storage medium carrying one or more sequences of instructions for reducing the recovery time after a failure, wherein execution of the one or more sequences of instructions by one or more processors causes the one or more processors to perform the steps of:

~~maintaining~~ storing a checkpoint value that indicates which records of a plurality of records have to be processed after the failure, wherein the plurality of records indicate changes for a plurality of data blocks; and

writing changes from volatile memory to nonvolatile memory to advance the checkpoint value based on a user-specified value that corresponds to how much work will be required during a redo phase of recovery.
11. (Currently Amended) The computer-readable storage medium of Claim 10, wherein execution of the one or more sequences of instructions by the one or more processors causes the one or more processors to further perform the step of:

maintaining, in volatile memory, one or more sorted buffer queues, wherein each sorted buffer queue includes queue entries that are inserted into said sorted buffer queue based on an index value associated with said queue entry,

wherein each queue entry reflects a change to a data block of the plurality of data blocks.
12. (Currently Amended) The computer-readable storage medium of Claim 11, wherein the one or more sorted buffer queues are one or more circular sorted buffer queues, and wherein a modulus operation is used to identify the index value associated with each circular sorted buffer queue entry when inserting a queue entry into the circular sorted buffer queue.

13. (Currently Amended) The computer-readable storage medium of Claim 11, wherein execution of the one or more sequences of instructions by the one or more processors causes the one or more processors to further perform the steps of:
maintaining a count of the queue entries in each of the one or more sorted buffer queues.
14. (Currently Amended) The computer-readable storage medium of Claim 11, wherein each of the one or more sorted buffer queues is associated with a target number of queue entries, and wherein execution of the one or more sequences of instructions by the one or more processors causes the one or more processors to further perform the steps of:
determining a count of the queue entries in each of the one or more sorted buffer queues;
and
if the count of the queue entries in a particular sorted buffer queue of the one or more sorted buffer queues is greater than target number of queue entries associated with the particular sorted buffer queue, then writing changes to a database to reduce the number of queue entries in the particular sorted buffer queue to the target number of queue entries associated with the particular sorted buffer queue.
15. (Currently Amended) The computer-readable storage medium of Claim 11, wherein the step of updating the checkpoint value comprises:
updating the checkpoint value to equal a byte offset in a redo log associated with the queue entry in the one or more sorted buffer queues that is associated with the least recently modified buffer in any queue entry in the one or more sorted buffer queues.
16. (Currently Amended) The computer-readable storage medium of Claim 10, wherein execution of the one or more sequences of instructions by the one or more processors causes the one or more processors to further perform the step of:
maintaining, in volatile memory, one or more partially sorted buffer queues, wherein each partially sorted buffer queue includes queue entries that are inserted into said

partially sorted buffer queue based on an index value associated with said queue entry,

wherein each queue entry reflects a change to a data block of the plurality of data blocks.

17. (Currently Amended) The computer-readable storage medium of Claim 10, wherein the checkpoint value comprises a byte offset to an identified redo log file.
18. (Currently Amended) A computer-readable storage medium carrying one or more sequences of instructions for controlling an amount of time that is needed to recover after the occurrence of a database system failure, wherein execution of the one or more sequences of instructions by one or more processors causes the one or more processors to perform the steps of:

maintaining storing a checkpoint value that indicates which records of a plurality of records have to be processed after the failure, wherein the plurality of records indicate changes for a plurality of data blocks;

determining a required recovery time, wherein the required recovery time indicates a maximum length of time that is to be allowed for recovering after said database system failure; and

writing changes from volatile memory to nonvolatile memory to advance the checkpoint value based on the maximum number of data block reads that can be performed in the required recovery time.